Maximal Adaptation, Minimal Guidance: Permissive Reactive Robot Task Planning with Humans in the Loop

Oz Gitelson¹, Satya Prakash Nayak², Ritam Raha², Anne-Kathrin Schmuck²

¹Yale University, USA.

oz.gitelson@yale.edu

²Max Planck Institute for Software Systems, Kaiserslautern, Germany.

sanayak, rraha, akschmuck@mpi-sws.org

Abstract—We present a novel framework for human-robot logical interaction that enables robots to reliably satisfy (infinite horizon) temporal logic tasks while effectively collaborating with humans who pursue independent and unknown tasks. The framework combines two key capabilities: (i) maximal adaptation enables the robot to adjust its strategy online to exploit human behavior for cooperation whenever possible, and (ii) minimal tunable feedback enables the robot to request cooperation by the human online only when necessary to guarantee progress. This balance minimizes human-robot interference, preserves human autonomy, and ensures persistent robot task satisfaction even under conflicting human goals. We validate the approach in a real-world block-manipulation task with a Franka Emika Panda robotic arm and in the Overcooked-AI benchmark, demonstrating that our method produces rich, emergent cooperative behaviors beyond the reach of existing approaches, while maintaining strong formal guarantees.

I. Introduction

Effective human-robot interaction (HRI) requires robots to operate alongside humans who pursue their own goals – often without explicitly revealing them. Such instances appear increasingly in domains ranging from smart manufacturing and logistics to assistive robotics in healthcare and households. In such scenarios, the robot must not only plan its moves to satisfy its own task but also adapt online to human behavior that may be cooperative, indifferent, or even obstructing its task. At the same time, HRI becomes more effective and enjoyable for the human, if robots do not only *react* to what humans do, but respect and even leverage human behavior rather than always overwriting or constraining it.

This paper addresses this challenge in the context of human-robot logical interactions (HR ℓ I) where the robot is assigned a high-level temporal task, expressed in linear temporal logic (LTL), and the human simultaneously pursues an unknown strategic latent task. As an example, consider the simplified manipulation task depicted in Fig. 1, where a Franka Emika Panda robotic arm takes turns with a human to place blocks in a 3×3 domain. The robots' task is to ensure that the majority of the cells is always eventually occupied with no adjacent cells filled, while the hidden human objective is to form a diagonal. If the domain of a logical task is restricted (as in this example), all possible strategic interactions of the human and the robot can be encoded in a two-player game graph, schematically illustrated in Fig. 2. Particular states in this graph fulfil the robots'

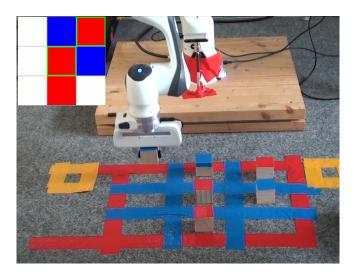


Fig. 1. A simplified gridworld block-manipulation domain from our experimental setup, where a Franka Emika Panda robotic arm takes turns with a human to place blocks in a 3×3 grid. The robot places blue blocks, while the human places red blocks. The top-left inset illustrates the robot's feedback to the human, suggesting the removal of the block in cell (1,3) or (2,2). A video of the experiment is available at https://youtu.be/61thSZDi5Ks.

(resp. humans') objective and are intended to be visited always again by the robot (resp. the human). However, as the robot and the human are both able to move blocks, they can obstruct the satisfaction of each others' goal. Importantly, this might already happen if both have the same goal, e.g., forming a diagonal. Here, the human can persist in forming the diagonal south-west-to-north-east, while the robot persist to form the south-east-to-north-west diagonal, resulting in a live lock. This problem is amplified if robot and human objectives differ, or are even incompatible.

To solve this problem, this paper introduces a novel $HR\ell I$ framework which enables robots to persistently satisfy their logical tasks, while ensuring human autonomy whenever possible and requesting cooperation only when necessary.

A. Related Work

Due to the enormous relevance of reliable human-robot interaction (HRI) for trustworthy autonomy, there is an

enormous body of work¹ in this research area. We therefore only focus on HR ℓ I scenarios described by a game between the robot and the human. This research line is rooted in the seminal papers [1], [2] which are motivated by the observation that LTL is a powerful specification language to describe strategic objectives such as traffic rules for autonomous driving [3], or robot navigation [4].

A natural formulation of $HR\ell I$ is as a two-player game between the robot and its environment. There exists a rich body of work on graph games [5]–[12] which can be leveraged to compute reactive strategies for the robot to fulfill diverse complex tasks. Unfortunately, most solutions typically either over-constrain the robot or the human. In the first case, a robot strategy is computed which ensures the robots' objective against all human strategies. This, however, makes solutions overly restrictive and typically (as in the manipulation domain example above) fails to produce a robot strategy altogether. In the second case, full cooperation between the robot and the human are assumed, and a strategy is computed for both. Thereby, the human is forced into a very rigid, fully constrained behavior – ensuring reliability but at the cost of suppressing human autonomy.

A middle-ground is provided by various approaches which allow for increased human autonomy. When only *logical safety* is concerned – i.e., ensuring that no *bad* strategic interaction happens between the human and the robot – reactive shielding mechanisms [13]–[16] which intervene with human behavior only to avoid such bad interactions, can be deployed. If, however, *logical liveness* objectives are present – i.e., requiring that something *good* (e.g., forming a diagonal in the above example) eventually happens – safety shielding is not sufficient to guarantee the satisfaction of the specification (as illustrated by the live-lock of human and robot trying to form a diagonal discussed before).

To mitigate these issues, many approaches explicitly model human behavior – either by predicting from trajectories [17] or by representing it as a Markov Decision Process [18] – and integrate these models into the synthesis framework. Alternatively, $HR\ell I$ can be directly modelled as a stochastic two-player game, as e.g. in [19]. While this introduces local viability of human strategies via stochasticity, it does not capture the need for strategic human autonomy.

To further improve human autonomy, admissibility-based methods, such as [20], [21] can be used, which enable robots to adopt behaviors that remain robust against a broad range of human actions while still ensuring task satisfaction. Orthogonal to this approach, Schuppe et al. [22] focus on interactive advice, where the robot provides assume–guarantee style guidance to the human to support the satisfaction of a *shared* objective with minimal cooperation by the human. However, these approaches require the robot to commit to a *fixed precomputed* strategy and rely either on stringent assumptions on human behavior, or on static, predefined forms of advice.

A different, but related approach to HR\ell only considers

a (non-reactive) logical *planning* objective for the robot and ensures safety of humans in its workspace by implementing the resulting plan of the robot via control-barrier functions (CBF), which act as a safety filter on the resulting underlying continuous robot dynamics [23]. This approach was recently incorporated into cooperative HRℓI frameworks, such as [24]–[26], where the (reactive) logical objectives of both the robot and the human are known, allowing an offline centralized game solution. Online human robot adaptations are only considered in the lower physical layer via CBFs but without any strategic autonomy. Similarly, recent dynamic game approaches to HRI, e.g. [27], [28] are focusing on the immediate physical, rather than the long-term strategic adaptation and interaction of humans and robots.

In contrast to these approaches, we present an autonomydriven approach to HR\ell I that focuses on high-level strategic interactions. While prior work with this focus, such as [20]– [22], [29], [30] result in pre-computed forms of cooperation or feedback and only consider fixed horizon objectives (as discussed before), our framework exploits the synergy between online adaptation and tunable feedback to generate complex emergent cooperation behavior for finite and infinite horizon LTL tasks. Formally, our novel framework is enabled by permissive strategy templates [31] for ω -regular games (derived from the LTL objective φ) which allow to concisely represent infinitely many strategies. Strategy templates have been applied to various contexts [32]-[39], but to the best of our knowledge, we are the first to apply them to $HR\ell I$. In particular, we leverage recent results from [33] which allow to capture all human and robot strategies that guarantee satisfaction of φ under minimal cooperation. This forms the theoretical basis for our novel online adaptation under strong formal guarantees.

B. Contribution

Our main contribution is a general framework for $HR\ell I$, which is applicable to the full class of LTL tasks and does not assume any particular strategic cooperation of the human. Conceptually, this framework ensures that the robot treats its interactions with a human not solely as a source of uncertainty to be constrained, but increasingly as a resource to be utilized. When pursuing a high-level LTL task φ , the robot strategy (i) adapts at runtime to the humans' (i.e., forming a south-west-to-north-east diagonal as pursued by the human instead of insisting on forming the other one) to maximize cooperation whenever possible, and (ii) provides strategic feedback to the human only when the robots' adaptation alone cannot ensure progress towards the satisfaction of φ (e.g., if the human persistently removes the middle block obscuring to form a diagonal the robot will ask the human to stop taking this move). This minimizes human-robot interference, maximizes cooperation and preserves human autonomy whenever possible, while still guaranteeing the eventual satisfaction of robot tasks if the human eventually listens to the provided feedback.

We validate our approach both in simulation and on robotic hardware. In addition to the robotic manipulation

¹Over two million results on Google Scholar searching for 'human robot interaction' on 15.9.25.

domain implementation in Fig. 1, we evaluate our framework in the *Overcooked-AI* simulation environment [40], a widely used benchmark for collaborative planning with multiple actors. In this domain, the human and the robot repeatedly perform cooking tasks with the objective of persistently producing soups. Each participant is assigned an independent LTL task that encodes a recipe specification. As in the gridworld domain, these tasks are private and not known to the other. The human behavior is simulated by a probabilistic strategy.

In this application scenario the advantage of using ω -regular specifications over commonly used reachability tasks becomes apparent. As both agents should produce as many specified soups as possible, they can actually cooperate even if their specifications are conflicting – simply by 'taking turns' in producing the 'right' soup. We show that this intuitive cooperative behavior autonomously emerges via the online adaptation and feedback mechanism provided by our framework. To the best of our knowledge, the complexity of the emergent $HR\ell I$ behavior provided by our framework thereby vastly exceeds the capabilities of all existing approaches while providing formal guarantees.

II. PROBLEM SETUP

We focus on a turn-based human-robot interaction scenario, where the robot and the human alternately act in a shared environment. Given a high-level temporal task for the robot, our goal is to develop a framework that enables the robot to adapt its strategy online to the human's observed behavior, and to provide feedback in a principled, tunable manner, so that the robot's task is reliably satisfied even when the human pursues an independent objective whose chosen strategy may conflict with the robot's progress.

A. Reactive Planning Domain

We model the interaction between the robot and the human as a reactive planning domain $\mathcal{D} = \langle S, s_0, A, AP, L \rangle$, where:

- $S = S_r \cup S_h$ is a set of states, partitioned into robot states S_r and human states S_h ;
- $s_0 \in S$ is the initial state;
- $A = A_r \cup A_h$ is the set of actions (modeled as directed edges), partitioned into robot actions $A_r \subseteq S_r \times S_h$ and human actions $A_h \subseteq S_h \times S_r$;
- AP is the set of task-related propositions that can either hold or not hold in a given state.
- $L\colon S\to 2^{AP}$ is a labeling function that labels each state with the set of propositions that hold in that state.

The planning domain can be specified using the planning domain description language (PDDL) [41], a standard language in the field of AI planning. In a PDDL description, a state captures relevant objects and their locations, while actions are defined in terms of preconditions and effects.

A run $\rho=s_0s_1s_2\ldots$ of the planning domain is an infinite sequence of states such that $s_0=s_0$ and for all $i\geq 0$, $(s_i,s_{i+1})\in A$, i.e., there is an action that takes the system from state s_i to state s_{i+1} . The run ρ induces a $trace\ L(\rho)=L(s_0)L(s_1)L(s_2)\ldots$, an infinite word over 2^{AP} , which is a

sequence of labels corresponding to the states in the run. We assume that the robot and human take turns to act, i.e., if $s_i \in S_r$, then $s_{i+1} \in S_h$, and vice versa.

A robot strategy $\pi_r\colon S^*S_r\to A_r$ is a function that maps a sequence of states (representing the history of the interaction) ending in a robot state to the action that the robot should take. A run $\rho=s_0s_1s_2\ldots$ is said to be π_r -run if for all $i\geq 0$, whenever $s_i\in S_r$, then $s_{i+1}=\pi_r(s_0s_1\ldots s_i)$. A human strategy π_h and π_h -run are defined similarly.

Example 1. Fig. 2 shows a partial view of a turn-based human-robot interaction modeled as a reactive planning domain. Each circle (e.g, $r0, r1, \ldots$) corresponds to a robot state in S_r and each rectangle (e.g., $h0, h1, \ldots$) corresponds to a human state in S_h . Inside each node, the 3×3 grid represents the environment: red squares denote human-placed objects, and blue circles denote robot-placed objects. Edges correspond to actions in A, alternating between human and robot moves according to the turn-based interaction. For example, the edge from state r2 to state h2 represents a robot action that removes a (blue circle) object from cell (1,1) (i.e., 1st row, 1st column). Fig. 1 shows an image of the same action being executed by a Franka robotic platform in a real-world setting.

A labeling function L can be defined to capture task-related propositions in AP. For instance, let us consider the propositions $AP = \{adj, diag, major\}$, where adjindicates that no two objects are adjacent (horizontally or vertically), diag indicates that a diagonal is fully occupied, and major indicates that at least 4 out of 9 cells are occupied (majority-occupied). In that case, adj holds only in states r0, t1, t2; diag holds in states r5, t2; and major holds in every state except h0 and r0. From the initial state h0, a possible run is $\rho = h0 (r1h1r2h2)^{\omega}$, which induces the trace $L(\rho) = \{\} \{major\}^{\omega}$.

B. Temporal Tasks as LTL formulas

To express robot/human tasks, we use *linear temporal* logic (LTL), a specification language that extends propositional logic with temporal operators [42]. Given a set AP of atomic propositions, LTL formulas are recursively defined as follows:

$$\varphi ::= \top \mid p \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \bigcirc \varphi \mid \varphi_1 \ U \ \varphi_2$$

where $p \in AP$ is an atomic proposition; \neg and \land are the boolean operators' negation and conjunction, respectively; and \bigcirc and U are the temporal operators 'next' and 'until', respectively. Other standard operators such as disjunction (\lor) , implication (\Rightarrow) , finally (\diamondsuit) , and globally (\Box) can be derived from the above operators. The semantics of LTL formulas are defined over infinite sequences of sets of atomic propositions in $(2^{AP})^{\omega}$ and can be found in standard books [43, Chapter 5.1.2]. We say a run ρ of a planning domain satisfies an LTL formula φ , denoted $\rho \models \varphi$, if the trace induced by ρ satisfies φ .

Example 2. Consider again the reactive planning domain in Ex. 1 with atomic propositions $AP = \{adj, diag, major\}$.

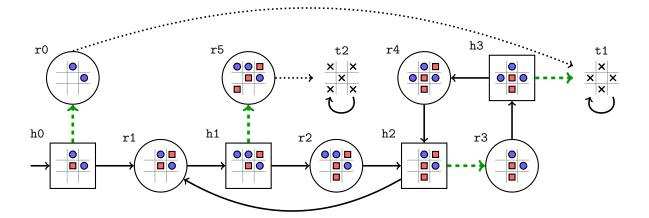


Fig. 2. An example of a partial reactive planning domain for turn-based human-robot interaction in a grid world. The robot controls the circle states, while the human controls the rectangle states. Each state contains a 3×3 grid showing the current positions of the human-placed objects (red squares) and the robot-placed objects (blue circles). Directed edges represent possible actions leading to successor states. The robot's objective is to repeatedly reach states with majority-occupied cells where the placed objects are non-adjacent (i.e., no two occupied cells are neighbors), as illustrated in states t1 and t2. Green dashed edges denote *live* actions, which are the suggested actions by the strategy templates, while dotted edges denote sequences of actions that lead to the target states.

Suppose the robot's task is to repeatedly reach states where no two objects are adjacent and at least four cells are occupied, which can be expressed by the LTL formula $\varphi_r = \Box \Diamond (adj \land major)$. A human's task could be to repeatedly reach a state where a diagonal is fully occupied, expressed by the LTL formula $\varphi_h = \Box \Diamond diag$. A run that repeatedly visits states t1 satisfies both tasks, while a run that eventually only loops in state t2 only satisfies the robot's task.

C. Problem Statement

In this work, we are interested in the scenario of humanrobot interaction where both the robot and the human have their own independent tasks, which are not known to each other. In such settings, each agent may follow a strategy that, even if unintentionally, can block or halt progress toward the other's task. Our goal is to develop a framework that enables the robot to adapt its strategy based on local observations and give feedback to the human when necessary, in order to persistently satisfy its own task over time.

Problem 1. Given a human-robot interaction modeled as a reactive planning domain \mathcal{D} with an LTL task φ for the robot in the presence of a human pursuing an unknown latent task, develop a framework that (a)

- 1) **captures** all cooperative behaviors of the human that enable the robot to satisfy its task φ ;
- 2) adapts the robot's strategy based on local observations of the human's strategic behavior during interaction;
- 3) **incorporates** a tunable mechanism for providing feedback to the human when needed, thereby facilitating the robot's progress toward fulfilling its task φ .

Example 3. Consider again the interaction in Ex. 1, where the robot's task is to repeatedly reach states with majority-occupied cells in which the placed objects are non-adjacent, as expressed by the LTL formula $\varphi_r = \Box \Diamond (adj \land major)$

in Ex. 2. Suppose the human, pursuing its own latent task, repeatedly places objects along a diagonal in the grid. From the robot's perspective, this behavior can lead to a run satisfying φ_r , and thus this (unintentional) cooperation of the human should be exploited by the robot to fulfill φ_r .

However, in order to do so, the robot can not commit to a single strategy in advance to ensure that φ_r is satisfied regardless of the human's behavior. To see this, let us assume that the robot choses to follow a fixed strategy which tries to satisfy φ_r via a configuration as in t1. If the human still attempts to form a diagonal, their interaction will get stuck in a cycle where task φ_r will never be satisfied. Instead, the robot must adapt its strategy by recognizing, from local observations, that the human is systematically filling the diagonal. It should then autonomously choose actions that lead to states where φ_r is satisfied via a configuration as in t2. This, however, is only possible because φ_r and the humans latent task can indeed be satisfied simultaneously.

If, however, both objectives are conflicting, it does not suffice for the robot to adapt to the human behavior. As an example, consider a scenario where the human persistently takes actions $h2 \rightarrow r3$ and $h3 \rightarrow r4$. In such cases, the robot must recognize that the human's behavior can no longer be exploited for (unintended) cooperation. The robot then gives feedback to the human—e.g., requesting to remove the object at cell (2,2) (by taking action $h3 \rightarrow t1$).

III. ADAPTABILITY AND FEEDBACK MECHANISMS VIA PERMISSIVE STRATEGY TEMPLATES

To reason about the strategic behavior of the robot for the LTL task, we reduce the planning domain along with the LTL task to a two-player game between the robot and human, as commonly done in the literature [43]. We then leverage the recently developed notion of *permissive strategy templates* [31], [32] in graph games to address Prob. 1.

A. ω-Regular Games

As a first step, we introduce the notion of two-player (turn-based) ω -regular games, which will serve as the foundation for our framework.

Definition 1. A two-player (turn-based) ω -regular game is a pair $\mathcal{G} = \langle \mathcal{D}, \Omega \rangle$, where $\mathcal{D} = \langle S, s_0, A, AP, L \rangle$ is a reactive planning domain (as defined in Sec. II-A), and $\Omega \subseteq S^{\omega}$ is an ω -regular set of infinite sequences of states that defines the winning condition of the game.

Such ω -regular games can be canonically represented as parity games [43]. A parity game is a special case of an ω -regular game where the winning condition $\Omega = Parity[c]$ is defined by a coloring function $c \colon S \to \mathbb{N}$ that assigns a natural number (color) to each state, and a run ρ belongs to Parity[c] if the maximum color that appears infinitely often in ρ is even. Using standard techniques [43], we can reduce a planning domain along with an LTL task to a parity game as formalized below.

Proposition 1. Given a reactive planning domain \mathcal{D} and an LTL formula φ over AP, we can construct a parity game $\mathcal{G} = \langle \mathcal{D}', \Omega \rangle$ such that there is a bijective correspondence between the runs of \mathcal{D} and the runs of \mathcal{D}' , and a run ρ of \mathcal{D} satisfies φ if and only if the corresponding run ρ' of \mathcal{D}' belongs to Ω .

B. Permissive Strategy Templates

In two-player games, a *strategy template* generalizes the notion of a strategy by succinctly representing an infinite family of strategies through local constraints on the agent's actions. Formally, a strategy template Π for agent i consists of the following types of constraints:

- Unsafe actions $U \subseteq A_i$: actions that the agent is prohibited from taking;
- Co-live actions C ⊆ A_i: actions that may only be taken finitely many times along any run;
- Live-groups $\mathcal{H} \subseteq 2^{A_i}$: sets of actions such that, if the source state of some $H \in \mathcal{H}$ is visited infinitely often, the agent must take at least one action from the set infinitely often.

A run ρ is said to comply with a strategy template Π , if it satisfies all the specified constraints. A strategy π is said to follow a strategy template Π , denoted $\pi \models \Pi$, if all π -runs comply with Π . For a detailed formal definition and further intuition, see [31].

Recent work [33] shows that, given a parity game between a robot and a human, it is possible to synthesize a strategy template for the human that captures all cooperative behaviors, and a corresponding strategy template for the robot that encompasses all strategies guaranteeing the winning condition against any cooperative human behavior.

Proposition 2. Given a parity game $\mathcal{G} = (\mathcal{D}, \Omega)$, a pair of strategy templates (Π_r, Π_h) for the robot and human, respectively, can be synthesized such that (i) every run $\rho \in \Omega$ also complies with Π_h ; and (ii) every strategy $\pi_r \models \Pi_r$ ensures that all π_r -runs complying with Π_h belong to Ω .

Example 4. For the grid world in Fig. 2 with the robot's task $\varphi_r = \Box \Diamond (adj \land major)$ as in Ex. 2, a parity game can be constructed as per Prop. 1 which has the same structure as the planning domain in Fig. 2 but with an appropriate coloring function c that assigns all states where adj \land major holds the color 2 (even) and all other states the color 1 (odd). This captures the robot's objective of repeatedly reaching states with majority-occupied cells where the placed objects are non-adjacent. Using the synthesis procedure in Prop. 2, we can compute a pair of strategy templates (Π_r, Π_h) for the robot and human, respectively, that capture cooperative behaviors. For instance, the human's strategy template Π_h includes live-groups $\{h3 \rightarrow t1\}$ (in addition to other live groups as shown by green dashed edges in Fig. 2), which ensure that the human does not consistently obstruct the robot's ability to make progress as discussed in Ex. 3. Similarly, the robot's strategy template Π_r includes live-groups that ensure the robot can always reach states satisfying adj∧major as long as the human follows Π_h .

C. Adaptation and Feedback Mechanism

While the results in Prop. 2 provide a foundation for capturing cooperative behaviors, addressing (1) of Prob. 1, they do not directly address the adaptation and feedback aspects outlined in (2) and (3) of Prob. 1. To this end, we propose a framework that leverages the permissive nature of strategy templates (Π_r, Π_h) to enable the robot to adapt its strategy based on its strategy template Π_r and provide feedback to the human based on the human's strategy template Π_h .

Adaptation: Since the template Π_r provides a set of possible actions at each state, the robot does not need to commit to a single strategy in advance. Instead, at runtime, the robot randomly selects an action from the set of enabled actions in Π_r at its current state. This approach allows the robot to adapt its choice of actions whenever it revisits a state. In particular, if an action taken by the robot does not lead to a desirable outcome (e.g., the human appears to be uncooperative), the run will eventually return to the same state, and by randomness, the robot can try different actions from the enabled set in Π_r . This adaptation mechanism effectively addresses (2) of Prob. 1.

Feedback Mechanism: To facilitate a tunable feedback mechanism as outlined in (3) of Prob. 1, we introduce a feedback threshold $\alpha \in [0,1]$ that determines how often the robot provides feedback to the human. Note that, since unsafe actions in Π_h are actions that the human must avoid satisfying the robot's task, the robot always communicates about unsafe actions whenever such an action is available at the current state. For other types of constraints in Π_h , i.e., co-live actions and live-groups, the robot observes the human's actions and monitors how often the human violates these constraints (i.e., takes co-live actions or avoids actions from a live-group). Whenever the frequency of such violations exceeds the feedback threshold α , the robot provides feedback to the human from the next time step onward until the frequency of violations drops below α . Due to

Prop. 2, as long as the human follows Π_h , the robot's strategy will ensure that the task is satisfied. This feedback mechanism enables the robot to provide feedback in a tunable manner, only when human actions deviate significantly from the cooperative behaviors captured by Π_h , thus effectively addressing (3) of Prob. 1.

Example 5. Continuing from Ex. 4, the robot can adapt its strategy at runtime by randomly selecting actions from the enabled set in its strategy template Π_r . For instance, suppose the human consistently places objects along the diagonal, as discussed in Ex. 3. Consider the scenario in Fig. 2 where the robot is at state h0, and currently it has made progress toward reaching a state with configuration as in t1. If the human continues to place objects along the diagonal, by taking actions h0 \rightarrow r1 (which violates the live-group {h0 \rightarrow r0} in Π_h), the robot will adapt its strategy that now may lead to states like t2, which can accommodate the human's diagonal placements while still satisfying φ_r . Now consider a scenario where the human, pursuing its own latent task, consistently places objects in a manner that obstructs the robot's task and loops between states h2, r3, h3, r4 as discussed in Ex. 3. In this case, once the frequency of violations of the live-groups exceeds the feedback threshold α , the robot will provide feedback to the human, requesting to take the live action h3 \rightarrow t1, which will help the robot make progress toward satisfying its task φ_r .

IV. EXPERIMENTS

We evaluate the proposed framework on two experimental domains that illustrate complementary aspects of our novel $HR\ell I$ framework. To demonstrate the power of our approach on a physical robotic platform, the first domain is the simplified gridworld block-manipulation setting depicted in Fig. 1 and described through our running example Ex. 1 to 5. This example provides an interpretable test bed for visualizing how the robot adapts its strategy online in response to human actions and issues feedback in a tunable manner once the human blocks progress beyond a specified threshold.

The second domain is the Overcooked-AI environment [40], a standard benchmark for collaborative planning, which highlights the power of guidance and adaptability to enable the fully autonomous emergence of complex strategic human robot interactions. In particular, the natural ω -regular structure of the specifications allows us to investigate different levels of difficulty for such emergent interactions, ranging from complete alignment to partial or full misalignment between the human's and the robot's tasks.

Together, these experiments showcase both the transparency of our method in symbolic domains and its power to produce complex emergent strategic human robot interactions fully autonomously at runtime, which go far beyond the capabilities of existing approaches.

Remark 1. Note that, [22] also considers a human-robot interaction 'Follow My Advice' (FMA) scenario where the robot provides advice to the human. However, their approach focuses on computing sufficient assumptions for the human

to device a static feedback mechanism based on a precomputed robot strategy to achieve a finite-horizon goal. Our framework, in contrast, emphasizes online adaptability of the robot's strategy to align with the human's behavior and employs a tunable feedback mechanism to handle persistent ω-regular objectives that we experimentally validate below. A faithful comparison would require re-formulating our settings to finite-horizon tasks, but this would undermine the core focus of our framework on the interplay between adaptability and feedback for persistent objectives.

A. Gridworld Block-Manipulation

In this experiment, we implemented the simplified gridworld block-manipulation domain on a Franka Emika Panda robotic arm running ROS jazzy to demonstrate the feasibility of our framework beyond the abstract model. The setup consists of a robot hand operating on a 3×3 workspace with tangible blocks that can be placed or removed, corresponding directly to the states illustrated in Ex. 1. The human interacts with the workspace by placing red blocks, while the robot places blue blocks. The system monitors the evolving configuration and evaluates whether the robot's task specifications (e.g., maintaining non-adjacent placements) are currently satisfied.

The reactive planning domain underlying this demonstration comprised approximately 7000 states and 18 propositions, encoding all possible placements of human and robot objects together with legal turn-taking moves. Our implementation required about 6 seconds to construct the parity game and synthesize the strategy templates, which was performed offline before execution. During execution, the robot follows its adaptive strategy: it updates its actions based on local observations of human moves, and when the human's behavior risks blocking task satisfaction (as in Ex. 3), the robot generates feedback through a display.

We include an image of the setup in Fig. 1 to illustrate how the abstract domain is realized in practice. This experiment highlights how our framework scales from the formal model to a real-world setting, providing an interpretable test bed to assess both adaptability and human feedback.

B. Overcooked-AI

We further evaluate our framework in the *Overcooked-AI* environment, a widely used benchmark for collaborative planning with multiple actors. In this domain, the human and the robot repeatedly perform cooking tasks with the objective of persistently producing soups. Each participant is assigned an independent LTL task that encodes a recipe specification. As in the gridworld domain, these tasks are private and not known to the other.

We consider three classes of experimental scenarios, characterized by the relation between the human's and the robot's recipes. In the first class, the recipes are identical, such that both participants unknowingly pursue the same task. In the second class, the recipes are distinct but compatible, meaning that there exists at least one type of soup that simultaneously satisfies both specifications. In the third class,

the recipes are incompatible, i.e., there is no soup that satisfies both specifications simultaneously. These classes capture increasing levels of misalignment between the tasks pursued by the human and the robot. Table I summarizes the recipe configurations considered in our experiments.

TABLE I
RECIPE CONFIGURATIONS IN THE OVERCOOKED-AI EXPERIMENTS.

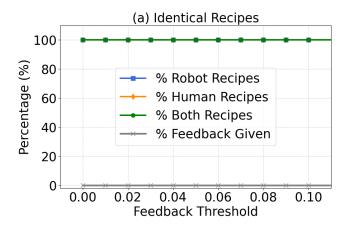
Scenario	Robot recipe	Human recipe
Identical	= 3 onions	= 3 onions
Incompatible	= 3 onions	= 2 onions
Compatible	≥ 2 onions	≤ 2 onions

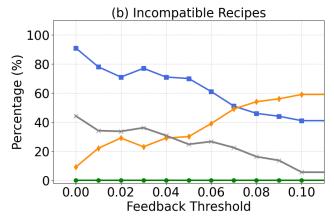
The Overcooked-AI environment provides a natural instantiation of persistent tasks, as both the human and the robot must repeatedly complete recipes over time. Each recipe specification corresponds to an ω -regular objective: an infinite run satisfies the task if the required recipe is produced infinitely often. This allows us to evaluate not only whether a single goal is reached, but also whether the tasks of the human and the robot can be persistently satisfied.

We model the Overcooked-AI environment as a reactive planning domain as per Sec. II-A, where the states encode the positions of the human and robot, the locations of ingredients, and the status of soups being cooked. The actions correspond to movement and interaction commands available to each participant. The ω -regular tasks for the human and the robot are specified as a parity condition over the states, which can be derived from the recipe specifications. The domain consists of approximately 68000 states with over 200 propositions encoding the relevant features of the environment. We implement the synthesis procedure in Prop. 2 to compute a pair of strategy templates (Π_r, Π_h) for the robot and human, respectively, that capture cooperative behaviors, which took around 3 minutes to compute offline before execution. The robot then executes the adaptation and feedback mechanism as described in Sec. III while the human is simulated by a probabilistic strategy for its recipe.

We ran the three recipe scenarios (identical, compatible, incompatible) under different values of the feedback threshold α ranging from 0.00 to 0.10. For each scenario, the system was executed until 10 soups were delivered. Each run lasted up to 500 moves with an execution time of 1 minute and was repeated 10 times to account for the randomness in the human's and robot's action selection. For each run, we recorded the following metrics: (i) the percentage of soups delivered that satisfy the robot's recipe, (ii) the percentage that satisfy both recipes simultaneously. Additionally, we tracked the frequency of feedback issued by the robot throughout the run. Fig. 3 presents the temporal evolution of these metrics across all scenarios, illustrating how adaptability and feedback influence persistent task satisfaction.

Identical recipes. In the identical recipe setting, the human and the robot unknowingly pursue the same recipe specification. As shown in Fig. 3 (a), both the human's and the robot's





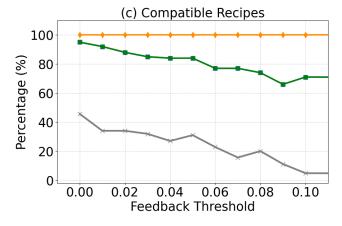


Fig. 3. Satisfaction of human and robot recipe tasks over time in Overcooked-AI for (a) identical recipes, (b) incompatible recipes, and (c) compatible recipes. The plots show the proportion of runs satisfying each objective as well as frequency of feedback given.

recipes are satisfied persistently across all runs. Crucially, no feedback is ever issued in this case. This illustrates the benefit of adaptability: even if the human and the robot start with different strategies for producing the same recipe, the robot is able to adjust online so that their behavior naturally aligns. In a system without adaptability, feedback would likely be needed to bring their strategies together, whereas our framework allows cooperation to emerge autonomously at runtime. Note that this also showcases the advantage of our framework over the static feedback mechanism in [22] (see

Rem. 1), which would issue feedback even when the human and robot have identical tasks, since it does not adapt the robot's strategy online.

Incompatible recipes. In the incompatible recipe setting, the human and the robot follow recipe specifications such that no soup can satisfy both recipes simultaneously. Consequently, as shown in Fig. 3 (b), the number of soups delivered that satisfy both recipes (green) remains zero. However, as objectives are formulated as ω -regular properties, it suffices to always eventually deliver a soup which satisfies the human's or robots' recipe, respectively. Hence, both agents can still cooperate by 'taking turns' in producing a soup via the humans' (orange) and the robots' (blue) recipe. Fig. 3 (b) shows that this intuitive cooperative behavior indeed autonomously emerges and is influenced by the feedback threshold. As the feedback threshold α increases, the robot becomes more lenient to human non-cooperation, leading to a higher fraction of runs in which the human's recipe is delivered while the robot's satisfaction decreases. This effect highlights the role of feedback tuning: with an appropriately chosen threshold (in this case, $\alpha = 0.07$), both the human and the robot succeed in satisfying their respective recipes approximately 50\% of the time persistently, demonstrating the importance of calibrated feedback sensitivity for the quality of the emerging cooperative behavior.

Compatible recipes. In the compatible recipe setting, the human and the robot pursue different recipe specifications, but there exists at least one soup that satisfies both. As shown in Fig. 3 (c), in our experiments the human persistently satisfies their recipe in all runs, while the robot adapts its strategy and issues feedback to ensure its own recipe is also satisfied. Even with higher thresholds, the robot manages to satisfy its recipe in more than 70% of runs by adapting its strategy online towards the shared goal. When combined with more frequent feedback (i.e., lower thresholds), the robot can further increase the rate of joint satisfaction, achieving up to 95% of runs where both recipes are satisfied. This demonstrates the synergy between adaptability and feedback tuning, which together enable persistent satisfaction of both recipes.

REFERENCES

- H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.
- [2] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, no. 7, pp. 1405– 1426, 2018.
- [3] N. Mehdipour, M. Althoff, R. D. Tebbens, and C. Belta, "Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges," *Automatica*, vol. 152, p. 110692, 2023. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/S0005109822005568
- [4] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for robots: Guarantees and feedback for robot behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 211–236, 2018.

- [5] R. Bloem, K. Chatterjee, and B. Jobstmann, "Graph games and reactive synthesis," in *Handbook of model checking*. Springer, 2018, pp. 921– 962.
- [6] M. Jurdzinski, "Deciding the winner in parity games is in UP \cap co-up," *Inf. Process. Lett.*, vol. 68, no. 3, pp. 119–124, 1998.
- [7] A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga, "Resource interfaces," in *EMSOFT*, ser. Lecture Notes in Computer Science, vol. 2855. Springer, 2003, pp. 117–133.
- [8] A. Ehrenfeucht and J. Mycielski, "Positional strategies for mean payoff games," *International Journal of Game Theory*, vol. 8, no. 2, pp. 109– 113, 1979.
- [9] L. Hélouët, N. Markey, and R. Raha, "Reachability games with relaxed energy constraints," *Information and Computation*, vol. 285, p. 104806, 2022. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/S089054012100122X
- [10] K. Chatterjee and L. Doyen, "Energy parity games," *Theoretical Computer Science*, vol. 458, pp. 49–60, 2012.
- [11] T. Banerjee, R. Majumdar, K. Mallik, A.-K. Schmuck, and S. Soudjani, "Fast symbolic algorithms for omega-regular games under strong transition fairness," *TheoretiCS*, vol. 2, 2023.
- [12] A. Anand, S. P. Nayak, R. Raha, I. Sağlam, and A.-K. Schmuck, "Fair quantitative games," in *International Conference on Foundations of Software Science and Computation Structures*. Springer Nature Switzerland Cham, 2025, pp. 331–354.
- [13] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, "Correcting robot plans with natural language feedback," in *Robotics: Science and Systems*, 2022.
- [14] F. A. Robinson, M. Velonaki, and O. Bown, "Smooth operator: Tuning robot perception through artificial movement sound," in *Proceedings* of the 2021 ACM/IEEE international conference on human-robot interaction, 2021, pp. 53–62.
- [15] S. Li, D. Park, Y. Sung, J. A. Shah, and N. Roy, "Reactive task and motion planning under temporal logic specifications," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 12618–12624.
- [16] D. Gundana and H. Kress-Gazit, "Event-based signal temporal logic tasks: Execution and feedback in complex environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10001–10008, 2022.
- [17] S. Junges, N. Jansen, J.-P. Katoen, U. Topcu, R. Zhang, and M. Hay-hoe, "Model checking for safe navigation among humans," in *International Conference on Quantitative Evaluation of Systems*. Springer, 2018, pp. 207–222.
- [18] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu, "Synthesis of human-in-the-loop control protocols for autonomous systems," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 450–462, 2016.
- [19] K. Muvvala, A. M. Wells, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Stochastic games for interactive manipulation domains," in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 2513–2519.
- [20] K. Muvvala, Q. H. Ho, and M. Lahijanian, "Beyond winning strategies: Admissible and admissible winning strategies for quantitative reachability games," *ICJAI*, 2025.
- [21] K. Muvvala and M. Lahijanian, "Admissibility over winning: A new approach to reactive synthesis in robotics," arXiv e-prints, pp. arXiv– 2410, 2024.
- [22] G. F. Schuppe, I. Torre, I. Leite, and J. Tumova, "Follow my advice: Assume-guarantee approach to task planning with human in the loop," in *Robotics: Science and Systems*, 2023.
- [23] K. Shi, J. Chang, S. Feng, Y. Fan, Z. Wei, and G. Hu, "Safe human dual-robot interaction based on control barrier functions and cooperation functions," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9581–9588, 2024.
- [24] V. Vasilopoulos, Y. Kantaros, G. J. Pappas, and D. E. Koditschek, "Reactive planning for mobile manipulation tasks in unexplored semantic environments," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 6385–6392.
- [25] Y. Zhang, V. N. Fernandez-Ayala, and D. V. Dimarogonas, "Multirobot human-in-the-loop control under spatiotemporal specifications," in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 4841–4847.
- [26] M. Srinivasan, S. Coogan, and M. Egerstedt, "Control of multi-agent systems with finite time control barrier certificates and temporal logic,"

- in 2018 IEEE Conference on Decision and Control (CDC), 2018, pp. 1991–1996.
- [27] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and systems*, vol. 2. Ann Arbor, MI, USA, 2016, pp. 1–9.
- [28] L. Peters, A. Bajcsy, C.-Y. Chiu, D. Fridovich-Keil, F. Laine, L. Ferranti, and J. Alonso-Mora, "Contingency games for multi-agent interaction," 2023.
- [29] Y. Huang, N. Xu, M. Guo, J. Li, and L. Shen, "Distributed and reactive controller synthesis for multi-agent systems under finite horizon temporal logic tasks," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 13485–13500, 2025.
- [30] Z. Zhou, S. Wang, Z. Chen, M. Cai, H. Wang, Z. Li, and Z. Kan, "Local observation based reactive temporal logic planning of human-robot systems," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 643–655, 2025.
- [31] A. Anand, S. P. Nayak, and A. Schmuck, "Synthesizing permissive winning strategy templates for parity games," in CAV (1), ser. Lecture Notes in Computer Science, vol. 13964. Springer, 2023, pp. 436–458.
- [32] A. Anand, K. Mallik, S. P. Nayak, and A. Schmuck, "Computing adequately permissive assumptions for synthesis," in *TACAS* (2), ser. Lecture Notes in Computer Science, vol. 13994. Springer, 2023, pp. 211–228.
- [33] A. Anand, A. Schmuck, and S. P. Nayak, "Contract-based distributed logical controller synthesis," in *Proceedings of the 27th ACM Inter*national Conference on Hybrid Systems: Computation and Control, HSCC 2024, Hong Kong SAR, China, May 14-16, 2024, E. Ábrahám and M. M. Jr., Eds. ACM, 2024, pp. 11:1–11:11.
- [34] S. P. Nayak and A. Schmuck, "Most general winning secure equilibria synthesis in graph games," in *TACAS (3)*, ser. Lecture Notes in Computer Science, vol. 14572. Springer, 2024, pp. 173–193.
- [35] A. Schmuck, P. Heim, R. Dimitrova, and S. P. Nayak, "Localized attractor computations for infinite-state games," in *CAV* (3), ser. Lecture Notes in Computer Science, vol. 14683. Springer, 2024, pp. 135–158.
- [36] A. Anand, S. P. Nayak, and A. Schmuck, "Strategy templates robust certified interfaces for interacting systems," in ATVA, ser. Lecture Notes in Computer Science, vol. 15054. Springer, 2024, pp. 22– 41
- [37] S. P. Nayak, L. N. Egidio, M. Della Rossa, A.-K. Schmuck, and R. M. Jungers, "Context-triggered abstraction-based control design," *IEEE Open Journal of Control Systems*, vol. 2, pp. 277–296, 2023.
- [38] A. Anand, S. P. Nayak, R. Raha, I. Saglam, and A. Schmuck, "Quantitative strategy templates," *CoRR*, vol. abs/2504.16528, 2025.
- [39] A. Anand, S. P. Nayak, R. Raha, and A. Schmuck, "Follow the stars: Dynamic ω -regular shielding of learned policies," *CoRR*, vol. abs/2505.14689, 2025.
- [40] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the utility of learning about humans for humanai coordination," *Advances in neural information processing systems*, vol. 32, 2019.
- [41] P. Haslum, N. Lipovetzky, D. Magazzeni, C. Muise, R. Brachman, F. Rossi, and P. Stone, An introduction to the planning domain definition language. Springer, 2019, vol. 13.
- [42] A. Pnueli, "The temporal logic of programs," in 18th annual symposium on foundations of computer science (sfcs 1977). ieee, 1977, pp. 46–57.
- [43] C. Baier and J.-P. Katoen, Principles of model checking. MIT press, 2008.